

### WELTORGANISATION FUR GEISTIGES EIGENTUM

Internationales Buro
Internationale Anmeldung Veröffentlicht nach dem Vertrag über die INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT)

(51) Internationale Patentklassifikation 6:

H03K 19/177, G06F 17/50

(11) Internationale Veröffentlichungsnummer: WO 98/31102

A1

(43) Internationales Veröffentlichungsdatum:

16. Juli 1998 (16.07.98)

(21) Internationales Aktenzeichen:

PCT/DE97/02999

(22) Internationales Anmeldedatum:

22. Dezember 1997 (22.12.97) (81) Bestimmungsstaaten: JP, europäisches Patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT,

(30) Prioritätsdaten:

196 54 593.5

DE 20. Dezember 1996 (20.12.96)

Veröffentlicht

Mit internationalem Recherchenbericht.

Vor Ablauf der für Änderungen der Ansprüche zugelassenen Frist. Veröffentlichung wird wiederholt falls Änderungen eintreffen.

(71) Anmelder: PACT INFORMATIONSTECHNOLOGIE GMBH [DE/DE]; Thelemannstrasse 15, D-81545 München (DE).

(72) Erfinder: VORBACH, Martin; Hagebuttenweg 36, D-76149 Karlsruhe (DE). MUNCH, Robert; Hagebuttenweg 36, D-76149 Karlsruhe (DE).

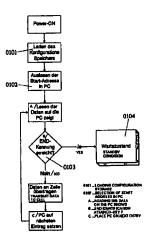
(74) Anwalt: ZAHN, Roland; Im Speitel 102, D-76229 Karlsruhe

(54) Title: RECONFIGURATION METHOD FOR PROGRAMMABLE COMPONENTS DURING RUNNING TIME

(54) Bezeichnung: UMKONFIGURIERUNGS-VERFAHREN FÜR PROGRAMMIERBARE BAUSTEINE ZUR LAUFZEIT

#### (57) Abstract

The invention relates to a method for reconfiguration during the running time of FPGA, in which there is a loading logic or several loading logics which react to signals of any kind and recognize and can process special loading logic commands within a configuration programme consisting of data and commands, and, on the basis of the source of an event, can compute an entry in a branch table. For this, there are one or more branch tables for locating the address of the configuration data to be loaded after computing. One or more configuration memory areas exist, in which one or more configuration programmes are loaded; and there are one or more FIFO memory areas into which configuration data is copied which could not be sent to the element or elements to be configured. When an event occurs, an address is computed in a branch table, based on the source of the event. FIFO memory area is provided and run through before each reloading, and, if the cell can not be reloaded, the configuration data is copied into it nearer the beginning; if the cell can be reloaded, the configuration data is transferred to the cell. The computed branch table entry is read-out, and the configuration data which is stored at the read-out address is loaded into the cell, or, if the cell cannot be reprogrammed, it is copied into the FIFO memory area.



#### (57) Zusammenfassung

In Verbindung mit einem Verfahren zum Umkonfigurieren zur Laufzeit von FPGA ist vorgesehen, dass eine Ladelogik oder mehrere Ladelogiken existieren, welche auf Signale, gleich welcher Art, reagieren und spezielle Ladelogik Befehle, innerhalb eines Konfigurationsprogramms, bestehend aus Daten und Befehlen, erkennen und verarbeiten können, sowie auf Grund der Quelle eines Ereignisses einen Eintrag in einer Sprung-Tabelle berechnen können. Dabei existieren eine oder mehrere Sprung-Tabellen zum Auffinden der Adresse der zu ladenden Konfigurationsdaten, welche berechnet wurde. Ein oder mehrere Konfigurations-Speicherbereiche existieren, in denen ein oder mehrere Konfigurationsprogramme geladen werden, und es existieren ein oder mehrere FIFO-Speicherbereiche, in die konfigurationsdaten kopiert werden, welche nicht an die oder das zu konfigurierende Element gesandt werden konnte. Trifft ein Ereignis ein, so wird auf Grund der Quelle des Ereignisses eine Adresse in einer Sprung-Tabelle berechnet. Es ist ein FIFO-Speicherbereich vorgesehen, der vor jedem Umladen durchlaufen wird, und falls die Zelle nicht umgelanden werden kann, in den die Konfigurationsdaten näher an den Anfang kopiert werden; falls die Zelle umgelanden werden kann, werden die Konfigurationsdaten an die Zelle übertragen. Der berechnete Sprung-Tabellen-Eintrag wird ausgelesen und die Konfigurationsdaten, welche an der ausgelesenen Adresse gespeichert sind, werden in die Zelle geladen, oder, falls die Zelle nicht umprogrammiert werden kann, in den FIFO-Speicherbereich kopiert.

### LEDIGLICH ZUR INFORMATION

Codes zur Identifizierung von PCT-Vertragsstaaten auf den Kopfbögen der Schriften, die internationale Anmeldungen gemäss dem PCT veröffentlichen.

AL	Albanien	ES	Spanien	LS	Lesotho	SI	Slowenien
AM	Armenien	FI	Finnland	LT	Litauen	SK	Slowakei
AT	Österreich	FR	Frankreich	LU	Luxemburg	SN	Senegal
AU	Australien	GA	Gabun	LV	Lettland	SZ	Swasiland
AZ	Aserbaidschan	GB	Vereinigtes Königreich	MC	Monaco	TD	Techad
BA	Bosnien-Herzegowina	GE	Georgien	MD	Republik Moldau	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagaskar	TJ	Tadschikistan
BE	Belgien	GN	Guinea	MK	Die ehemalige jugoslawische	TM	Turkmenistan
BF	Burkina Faso	GR	Griechenland		Republik Mazedonien	TR	Türkei
BG	Bulgarien	HU	Ungam	ML	Mali	TT	Trinidad und Tobago
BJ	Benin	Œ	Irland	MN	Mongolei	UA	Ukraine
BR	Brasilien	IL	Israel	MR	Mauretanien	UG	Uganda
BY	Belarus	IS	Island	MW	Malawi	US	Vereinigte Staaten von
CA	Kanada	IT	Italien	MX	Mexiko		Amerika
CF	Zentralafrikanische Republik	JP	Japan	NE	Niger	UZ	Usbekistan
CG	Kongo	KE	Kenia	NL	Niederlande	VN	Vietnam
CH	Schweiz	KG	Kirgisistan	NO	Norwegen	YU	Jugosławien
CI	Côte d'Ivoire	KP	Demokratische Volksrepublik	NZ	Neuseeland	zw	Zimbabwe
CM	Kamerun		Korea	PL	Polen		
CN	China	KR	Republik Korea	PT	Portugal		
CU	Kuba	KZ	Kasachstan	RO	Rumänien		
CZ	Tschechische Republik	LC	St. Lucia	RU	Russische Föderation		
DE	Deutschland	LI	Liechtenstein	SD	Sudan		
DK	Dänemark	LK	Sri Lanka	SE	Schweden		
EE	Estland	LR	Liberia	SG	Singapur		

Umkonfigurierungs-Verfahren für programmierbare Bausteine zur Laufzeit

Hintergrund der Erfindung Stand der Technik Programmierbare Bausteine mit zwei oder mehrdimensionaler Zellanordnung (insbesondere FPGAs, DPGAs und DFPs o.ä.) werden heutzutage auf zwei verschiedene Arten programmiert.

- 1. Einmalig, das heißt die Konfiguration kann nach der Programmierung nicht mehr geändert werden. Alle Konfigurierten Elemente des Bausteins führen also die gleiche Funktion, über den gesamten Zeitraum durch, in dem die Anwendung abläuft.
- 2. Im Betrieb, das heißt die Konfiguration kann nach Einbau des Bausteins, durch daß Laden einer Konfigurationsdatei, zum Startbeginn der Anwendung, geändert werden. Die meisten Bausteine (insbesondere die FPGA Bausteine), lassen sich während des Betriebes nicht wieder umkonfigurieren. Bei umkonfigurierbaren Bausteinen, ist eine Weiterverarbeitung von Daten während des Umkonfigurierens meistens nicht möglich und die benötigte Umkonfigurierungszeit erheblich zu groß.

Neben den FPGAs, gibt es noch die sogenannten DPGAs. Diese Bausteine speichern eine kleine Anzahl an verschiedenen Konfigurationen, welche durch ein spezielles Datenpaket ausgewählt werden. Eine Umkonfigurierung dieser Speicher während der Laufzeit ist nicht möglich.

#### Probleme

Große Probleme bereitet die Umkonfigurierung von gesamten programmierbaren Bausteinen oder Teilen davon während der

Laufzeit und dabei besonders die Synchronisation. Alle bisherigen Lösungen halten die Verarbeitung des kompletten Bausteins, während der Umkonfigurierung, an. Ein weiteres Problem ist die Selektion der neu zu ladenden Teilkonfiguration und das Integrieren dieser Teilkonfiguration in die bereits Bestehende Konfiguration.

Verbesserung durch die Erfindung
Durch das in der Erfindung beschriebene Verfahren ist es
möglich einen, zur Laufzeit, umkonfigurierbaren Baustein,
effizient und ohne Beeinflussung der nicht an der
Umkonfigurierung beteiligten Bereiche, umzuladen. Weiterhin
ermöglicht das Verfahren das Selektieren von Konfigurationen
in Abhängigkeit der aktuellen Konfiguration. Das Problem der
Synchronisation der an der Umkonfiguration beteiligten und
nicht beteiligten Bereiche wird ebenfalls gelöst.

Beschreibung der Erfindung, Abstrakt
Beschrieben wird ein Verfahren zum Umkonfigurieren von
programmierbaren Bausteinen, bestehend aus einer zwei oder
mehrdimensionalen Zellanordnung. Das Verfahren ermöglicht die
Umkonfigurierung des Bausteins oder der Bausteine, ohne die
Arbeitsfähigkeit, der nicht an der Umkonfigurierung
beteiligten Zellen, einzuschränken. Das Verfahren ermöglicht
das Laden von kompletten Konfigurationen oder von
Teilkonfigurationen in den oder die programmierbaren
Bausteine. Die Einzelheiten und besondere Ausgestaltungen,
sowie Merkmale des erfindungsgemäßen Verfahrens zum
Umkonfigurierens von programmierbaren Bausteinen, sind
Gegenstand der Patentansprüche.

Detailbeschreibung der Erfindung

Das beschriebene Verfahren setzt einen programmierbaren Baustein voraus, welcher folgende Eigenschaften aufweist:

1. Ladelogik

Die Ladelogik ist der Teil des Bausteins, welcher das Laden und Eintragen von Konfigurationsworten in die zu konfigurierenden Elemente des Bausteins (Zellen) durchführt.

2. Zellen

Der Baustein besitzt eine Vielzahl an Zellen, welche einzeln durch die Ladelogik adressiert werden können.

3. Rückmeldung Ladelogik

Jede Zelle oder Gruppe von Zellen muß der Ladelogik mitteilen können, ob sie umkonfiguriert werden kann.

4. Rückmeldung Zellen

Jede Zelle muß die Möglichkeit haben, ein STOP Signal an die Zellen zu senden, von denen sie ihre zu verarbeitenden Daten erhält.

5. START/STOP Kennung

Jede Zelle muß eine Möglichkeit besitzen, ein START/STOP Kennung einzustellen.

- a. Die START Kennung zeichnet eine Zelle als den Beginn einer längeren Verarbeitungskette (Makro) aus.
- b. Die STOP Kennung markiert das Ende des Makros, also den Punkt, an dem die Verarbeitung des Makros ein Ergebnis geliefert hat.

Aufbau eines Konfigurationswortes
Die Ladelogik ist eine Zustandsmaschine, welche
Konfigurationsworte verarbeiten kann. Neben
Konfigurationsworten für Zellen, existieren Einträge, welche
durch die Ladelogik als Befehle erkannt werden können. Es ist
also möglich zu unterscheiden ob der Inhalt des

Konfigurationswortes an eine Zelle zu übertragen ist oder einen Befehl für die Zustandsmaschine darstellt. Ein Konfigurationswort, welches an Zellen des Bausteins übertragen wird, muß dabei mindestens folgende Daten enthalten:

- 1. Adresse der Zelle. Zum Beispiel als lineare Nummer oder als X.Y Koordinaten.
- 2. Konfigurationswort, welches in die Zelle übertragen wird. Kennungen und Befehle für die Ladelogik

Für eine korrekte Arbeitsweise der Ladelogik, muß diese nur zwei Befehlsworte erkennen können. Dies sind:

#### 1. END

Dies ist ein Befehl, welche die Ladelogik in einen Zustand versetzt, in dem sie auf das Eintreffen von Ereignissen von Zellen, wartet. (Figur 2)

2. DISPATCH(Eintragsnummer, Adresse)
Die Ladelogik trägt in die Adresse, welche durch den
Parameter Eintragsnummer angegeben wird, der Sprung-Tabelle
den Wert des Parameters Adresse ein.

Weiterhin kann die Ladelogik einen Eintrag als Leer-Eintrag erkennen. Dies wird dadruch erreicht, daß ein bestimmtes Bit-Muster als Leer-Kennung definiert ist, welches durch die Ladelogik erkannt werden kann.

### Die Sprung-Tabelle

Im Konfigurationsspeicher, existiert eine Sprung-Tabelle (0506). Die Größe der Sprung-Tabelle ist dabei so gewählt, daß für jede Zelle, welche von der Ladelogik addresiert werden kann, genau ein einziger Eintrag vorhanden ist. Zu jeder Zelladresse existiert genau ein einziger Eintrag in der Sprung-Tabelle, welcher durch die Ladelogik berechnet werden kann. (Figur 5 und 6)

In einem Eintrag der Sprung-Tabelle steht eine Speicheradresse (0601). Diese Speicheradresse gibt an, von wo weitere Konfigurationsdaten (0508), aus dem Konfigurationsspeicher, zu lesen sind, falls von dieser Zelle eine Rückmeldung an die Ladelogik erfolgt.

#### Start des Systems

Durch einen Reset, also dem Rücksetzen des Systems, beginnt die Ladelogik mit dem Empfangen oder Laden von Konfigurationsdaten, von einem Speicher, in den Konfigurationsspeicher (0101). Alle Zellen des Bausteines sind in dem Zustand, in dem sie konfiguriert werden können. Danach springt die Ladelogik, durch Laden des Programmzählers (PC) (0505), an eine Speicherstelle, welche die Adresse einer Startkonfiguration (0507) enthält (0102). Diese Startkonfiguration wird solange abgearbeitet, bis die Ladelogik eine END-Kennung erkennt (0103). Diese Startkonfiguration programmiert den Baustein derart, daß eine Verarbeitung von Daten beginnen kann. Nach dem Eintragen der Startkonfiguration, wechselt die Ladelogik, auf Grund der END-Kennung, in einen Zustand, in dem sie auf Ereignisse von den Zellen wartet (0104).

Eintreffen eines Ereignisses einer Zelle
Nach der Verarbeitung von Daten kann eine Zelle eine
Rückmeldung an die Ladelogik senden. Diese Rückmeldung
(Ereignis) zeigt an, daß die Zelle und damit das Makro in dem
die Zelle enthalten ist, seine Arbeit beendet hat und das
Umladen erfolgen kann.

Bevor allerdings mit dem Laden einer neuen Konfiguration begonnen wird, wird der nachfolgend beschriebene FIFO-Speicher (First-In-First-Out Speicher) abgearbeitet (0201).

Wichtig ist, daß der Speicher als FIFO-Speicher organisiert ist. Diese Organisation garantiert, daß Zellen die im ersten Versuch nicht umgeladen werden konnten, garantiert im zweiten Versuch als erste wieder an der Reihe sind. Dadurch wird verhindert, daß Zellen welche zwischenzeitlich signalisiert haben, daß sie umkonfiguriert werden können, ganz nach hinten in der Bearbeitung rutschen. In diesem Fall könnte ein Deadlock-Situation auftreten, in der das eine Makro erst umkonfiguriert werden kann, wenn ein anderes Makro umkonfiguriert wurde.

Durch die Rückmeldung an die Ladelogik, erhält die Ladelogik auch die Adresse oder Nummer der Zelle, welche die Rückmeldung ausgelöst hat. Mit Hilfe dieser Nummer, wird der passende Eintrag in der Sprung-Tabelle selektiert (0203, 0204). Die Adresse, welche in diesem Eintrag enthalten ist, gibt den Beginn der zu ladenden Konfiguration innerhalb des Konfigurationsspeichers an (0205).

#### FIFO-Speicher

Das Verfahren muß berücksichtigen, daß es sein kann, daß einige Zellen ihre Arbeit noch nicht beendet haben, diese Zellen jedoch schon umgeladen werden sollen. Alle Konfigurationsdaten der Zellen, bei denen eine solche Bedingung zu trifft, werden in einen speziellen Speicherbereich (FIFO-Speicher) kopiert (0506).

Jedesmal, bevor eine neue Konfiguration geladen werden soll, wird der FIFO-Speicher durchlaufen. Da eine neue Konfiguration geladen werden soll, haben einige Zellen ihre Arbeit beendet und sind in den Zustand 'umkonfigurierbar' übergegangen. Unter diesen Zellen können sich auch solche befinden, bei denen eine Umkonfigurierung, durch die Ladelogik, in einem früheren Versuch gescheitert ist, da diese Zellen ihre Arbeit noch nicht beendet hatte, diese

Umkonfigurierung jetzt aber erfolgreich durchgeführt werden kann.

Die Ladelogik lädt den PC mit dem Inhalt des Registers, welches auf den Beginn des FIFO-Speicher zeigt (FIFO-Start-REG) (0502) und ließt die Daten aus dem FIFO-Speicher. Ein Vergleich stellt fest, ob das Ende des FIFO-Speichers erreicht wurde (0301). Ist dies der Fall, so wird an die Stelle in der Zustandsmaschine zurückgesprungen, an der die Umkonfigurierung fortläuft (0202).

Die Abarbeitung des FIFO-Speichers geschieht ähnlich dem einer Konfiguration innerhalb des Konfigurationsspeichers. Es kann der Fall eintreten, daß eine Zelle auch bei einem weiteren Versuch immer noch nicht umkonfiguriert werden kann. In diesem Fall werden die Konfigurationsdaten, falls eine leere Speicherstelle weiter vorne im FIFO-Speicher existiert, in diese Speicherstelle kopiert (0302).

Dieser Kopiervorgang wird dadruch erreicht, daß die Ladelogik die Startadresse des FIFO-Speichers im FIFO-Start-REG (0502) gespeichert hat und die Endeadresse im FIFO-End-REG (0503). Weiterhin kennt die Ladelogik die Adresse des nächsten freien Eintrags (beginnend vom Anfang des FIFO-Speichers) mittels des FIFO-Free-Entry-REG (0504, 0303). Nachdem das Konfigurationswort in den freien Eintrag kopiert wurde (0304), positioniert die Ladelogik den Zeiger des FIFO-Free-Entry-REG auf den nächsten freien Eintrag (0305), innerhalb des FIFO-Speichers. Die Suche erfolgt dabei in Richtung des Endes des FIFO-Speichers. Danach wird der PC auf den nächsten Eintrag innerhalb des FIFO-Speichers gesetzt (0306).

#### Umladen von Zellen

Die Ladelogik liest nun die Konfigurationsdaten aus dem Konfigurationsspeicher. In diesen Daten ist die Adresse der Zelle enthalten, welche umgeladen werden soll (Figur 4). Jede

Zelle kann signalisieren, daß sie umgeladen werden kann. Die Ladelogik testet dies (0401). Kann die Zelle umgeladen werden, werden die Konfigurationsdaten von der Ladelogik an die Zelle übertragen.

Ist die Zelle noch nicht bereit, werden die durch die Ladelogik gelesenen Daten in einen Speicherbereich, den FIFO-Speicher, innerhalb des Konfigurationsspeichers geschrieben (0402). Die Adresse an welche die Daten geschrieben werden, ist in einem Register, innerhalb der Ladelogik, abgelegt (FIFO-End-Reg) (0503).

Dieser Vorgang wird so oft wiederholt, bis die Ladelogik die END-Kennung des Konfigurationsprogramms erkennt und wieder in den Zustand übergeht, in dem die Ladelogik auf Ereignisse der Zellen wartet (0403).

#### Aufbau des Konfigurationsprogramms

Nachdem eine Zelle das Signal zum Umladen gegeben hat und das Makro, in dem die Zelle integriert ist, umgeladen wurde, entsteht eine neue Konfiguration. Die Zelle die vorher das Signal an die Ladelogik geben hat, kann jetzt eine ganz andere Aufgabe haben, insbesondere kann sie nicht mehr die Zelle sein, welche ein Umladesignal, an die Ladelogik abschickt. Wobei es möglich sein kann, daß in der neuen Konfiguration wieder die selbe Zelle das Umladesignal an die Ladelogik schickt.

Mittels des DISPATCH-Befehls, innerhalb des Konfigurationsprogramms, kann eine neue Adresse an die Eintragsposition der Zelle in der Sprung-Tabelle geschrieben werden (0604). Diese neue Adresse kann auf eine neue Konfiguration oder Teilkonfiguration zeigen, welche bei einer Rückmeldung von dieser Zelle geladen werden soll.

Kurzbeschreibung der Diagramme

Fig. 1 ist ein Ablaufplan der Schritte, die nach einem Systemstart durchzuführen sind.

Fig. 2 ist ein Ablaufplan der Schritte, die nach dem Eintreffen einer Umkonfigurierungsanforderung durchzuführen sind.

Fig. 3 ist ein Ablaufplan der Schritte, die bei der FIFO-Speicher Bearbeitung durchzuführen sind.

Fig. 4 ist ein Ablaufplan der Schritte, die bei der Konfigurierung der Zellen durchzuführen sind.

Fig. 5 zeigt die Ladelogik mit ihren Registern. Weiterhin ist der Konfigurationsspeicher sowie die Unterteilung in Sprung-Tabelle, Start-Konfiguration, weitere Konfigurationen und der FIFO-Speicher zu sehen.

Fig. 6 zeigt zwei Ausschnitte aus einem Konfigurationsprogramm und vier Ausschnitte aus der Sprung-Tabelle und wie diese in zeitlichem Zusammenhang stehen.

### Detailbeschreibung der Diagramme

Figur 1 zeigt in einem Ablaufplan, welche Schritte nach einem Systemstart durchzuführen sind. Durch einen Vergleich mit der END-Kennung der Start-Konfiguration wird in den Wartezustand gesprungen (0104).

Figur 2 zeigt in einem Ablaufplan die notwendigen Schritte, welche während des Wartezustandes und, nach dem eine Umkonfigurierung durch eine Zelle signalisiert wurde, durchzuführen sind. Der Ablaufplan besitzt einen Einsprungspunkt (0202), der von anderer Stelle angesprungen wird.

Figur 3 zeigt in einem Ablaufplan, wie die Behandlung

des FIFO-Speichers durchzuführen ist. Weiterhin ist dargestellt, wie der Kopiervorgang innerhalb des FIFO-Speichers arbeitet.

Figur 4 zeigt in einem Ablaufplan, welche Schritte bei der Umkonfigurierung der Zellen notwendig sind und wie eine Konfiguration innerhalb des Konfigurierungsprogramms abgearbeitet wird.

Figur 5 stellt die Ladelogik und ihre Register dar. Die Ladelogik besitzt fünf verschiedene Register. Dies sind:

1. Das Start-Konfiguration-REG (0501). In diesem Register steht die Adresse der Startkonfiguration innerhalb des Konfigurationspeichers. Die Daten sind derart in dem Konfigurationsprogramm enthalten, daß sie von der Ladelogik erkannt und in das Start-Konfiguration-REG übernommen werden können.

- 2. Ein FIFO-Start-REG (0502). Das FIFO-Start-REG zeigt auf den Beginn des FIFO-Speicherbereichs, innerhalb des Konfigurationsspeichers.
- 3. Ein FIFO-End-REG (0503). Das FIFO-End-REG kennzeichnet das Ende des FIFO-Speichers. An diese Stelle werden die Konfigurationsworte kopiert, welche durch die Ladelogik nicht verarbeitet werden konnten.
- 4. Ein FIFO-Free-Entry-REG (0504). Das FIFO-Free-Entry-REG zeigt auf den freien Eintrag, der dem Beginn (FIFO-Start-REG) des FIFO-Speichers am nächsten ist. An diese Stelle werden die Konfigurationsworte kopiert, welche während des Durchlaufens des FIFO-Speichers, wiederum nicht durch die Ladelogik verarbeitet werden konnten.
- 5. Einen Programmzähler (PC). Der PC zeigt auf die Adresse, innerhalb des Konfigurationsspeichers, in dem das nächste, durch die Ladelogik zu verarbeitende, Konfigurationswort, steht.

6. Ein Adress-REG (0510). In diesem Register wird die Adresse einer zu addressierenden Zelle gespeichert.

- 7. Ein Data-REG (0511). Dieses Register speichert die Konfigurationsdaten, welche an die Zelle gesendet werden sollten, welche durch das Adress-REG angesprochen wird.
- 8. Ein Dispatch-REG (0512). Das Dispatch-REG speichert die Adresse des Eintrags in der Sprung-Tabelle, auf welchen die Ladelogik zugreift.

Weiterhin ist der Konfigurationsspeicher und sein verschiedenen Sektionen zu sehen. Dies sind:

- 1. Die Sprung-Tabelle (0506). Für jede Zelle, welche durch die Ladelogik konfigurierbar ist, existiert ein einziger Eintrag. In diesem Eintrag steht die Adresse, welche bei einer Signalisierung durch diese Zelle, in den PC geladen wird.
- 2. Eine Start-Konfiguration (0507). Die Start-Konfiguration ist jeden Konfiguration, welche nach dem starten des Systems in den Baustein geladen wird.
- 3. Weitere Konfigurationen (0508). Diese Konfigurationen können während der Laufzeit des Systems in den Baustein geladen werden. Die Konfigurationen bestehen aus Konfigurationswörtern und Ladelogik Befehlen.
- 4. Einen FIFO-Speicher Bereich (0509). Der FIFO-Speicher Bereich enthält alle die Konfigurationsworte, welche durch die Ladelogik in einem ersten Versuch nicht erfolgreich verarbeitet werden konnten.

Figur 6 zeigt zwei Ausschnitte aus einer Konfiguration. In diesen Ausschnitten sind die Befehle und Konfigurationsworte zu sehen, welche durch die Ladelogik, verarbeitet werden. Weiterhin sind zwei Ausschnitte aus der Sprung-Tabelle zu sehen (0601 und 0607) und der Zustand dieser Ausschnitte

(0602 und 0608) nach der Abarbeitung der beiden Konfigurationsausschnitte.

#### Ausführungsbeispiele

Es wird angenommen, daß ein Baustein oder mehrere Bausteine durch eine Ladelogik, wie in der Erfindung beschrieben, umkonfiguriert werden sollen. Weiterhin sei angenommen, daß das System bereits die Startkonfiguration geladen hat, und die Ladelogik sich im Zustand 'warten auf ein Ereignis' befindet. Die Ausführung beginnt mit dem Eintreffen eines Ereignisses von Zelle Nummer 41.

Die Ladelogik beginnt zuerst mit der Abarbeitung des FIFO-Speichers (0201). Dabei wird der Beginn des FIFO-Speichers aus dem Register FIFO-Start-REG in den PC übertragen. Die Daten an der Stelle, auf die der PC zeigt, werden gelesen. Nun wird überprüft, ob das Ende des FIFO-Speichers erreicht wurde. Dies ist in diesem Ausführungsbeispiel der Fall, da das System das erste Mal umgeladen wird.

Die Adresse der Zelle, welche das Signal ausgelöst hat, wird durch die Ladelogik in eine Adresse der Sprung-Tabelle umgerechnet. Diese berechnete Adresse wird in das Distpatch-REG geladen (0512). Die Ladelogik liest nun die Adresse aus der Sprung-Tabelle (0506), welche an der Speicheradresse gespeichert ist, die durch das Dispatch-REG adressiert wird (0601). Diese Adresse wird in den PC geladen.

Daraufhin beginnt die Verarbeitung der Konfigurationsworte (0603). Es sei angenommen, daß der Befehl Nummer 3 (1,3 MUL) nicht ausgeführt werden kann, da die Zelle mit der Adresse (1,3) nicht umkonfiguriert werden kann. Die Daten werden nun in den FIFO-Speicher kopiert. Mit Erreichen des DISPATCH-Befehls (0604) wird an die Adresse 41 in der Sprung-Tabelle, eine neue Adresse eingetragen (0602). Der END-Befehl versetzt

die Ladelogik wieder in den 'warten auf ein Ereignis' Zustand.

Nach einiger Zeit triff nun wieder ein Signal von der Zelle 41 ein. Jetzt steht an der Adresse 42 der Sprung-Tabelle eine andere Adresse (0602). Die Ladelogik arbeitet wieder zuerst den FIFO Speicher ab. Nun befinden sich Daten in dem FIFO-Speicher.

Die Daten aus dem FIFO-Speicher werden gelesen und es wird versucht die adressierte Zelle mit den Daten zu laden. Da die Zelle jetzt umkonfiguriert werden kann, gelingt dies. Der Eintrag des FIFO Speichers wird daraufhin mit einer Leer-Kennung beschrieben.

Die ursprüngliche Verarbeitung wird fortgeführt und das Lesen von Konfigurationsdaten beginnt nun an einer unterschiedlichen Adresse (0605).

Diese Konfiguration wird abgearbeitet, der DISPATCH-Befehl schreibt diesmal eine Adresse in den Eintrag Nummer 12 der Sprung-Tabelle (0606). Danach versetzt der END-Befehl die Ladelogik wieder in den Zustand 'warten auf ein Ereignis'. Dieses Wechselspiel wiederholt sich während der gesamten Laufzeit des Systems.

Begriffsdefinition

konfigurierbares Element

Stellt eine Einheit eines Logik-Bausteines dar, welche durch
ein Konfigurationswort für eine spezielle Funktion
eingestellt werden kann. Konfigurierbare Elemente sind somit,
alle Arten von RAM-Zellen, Multiplexer, Arithmetische
logische Einheiten, Register und alle Arten von interner und
externer Vernetzungsbeschreibung etc.

Konfigurieren Einstellen der Funktion und Vernetzung eines konfigurierbaren Elements.

Konfigurationsspeicher Der Konfigurationspeicher enthält ein oder mehrere Konfigurationsworte.

Konfigurationswort Ein Konfigurationswort besteht aus einer beliebig langen Bit-Reihe. Diese Bit-Reihe stellt eine gültige Einstellung für das zu konfigurierende Element dar, so das eine funktionsfähige Einheit entsteht.

Ladelogik Einheit zum Konfigurieren und Umkonfigurieren von programmierbaren Bausteinen. Ausgestaltet durch einen speziell an seine Aufgabe angepaßten Mikrokontroller oder eine Zustandsmaschine.

Makro Ein Makro ist eine Menge von Zellen, welche zusammen eine Aufgabe, Funktion etc. implementieren.}

Umkonfigurieren Neues Konfigurieren von einer beliebigen Menge von konfigurierbaren Elementen eines programmierbaren Bausteins während eine beliebige Restmenge von

konfigurierbaren Elementen ihre eigenen Funktionen fortsetzen (vgl. konfigurieren).

Rückmeldung Eine Rückmeldung ist eine Aktion, welche eine Zelle auslösen kann. Bei einer Rückmeldung können verschiedene Informationen an die Einheit gesandt werden, welche die Rückmeldung empfängt.

Zelle siehe konfigurierbares Element.

Zustandsmaschine Logik, die diversen Zuständen annehmen kann. Die Übergänge zwischen den Zuständen sind von verschiedenen Eingangsparametern abhängig. Diese Maschinen werden zur Steuerung komplexer Funktionen eingesetzt und entsprechen dem Stand der Technik

#### Patentansprüche

1. Verfahren zum Umkonfigurieren zur Laufzeit von programmierbaren Bausteinen, mit einer zwei oder mehrdimensionalen Zellanordnung (zum Beispiel FPGAs, DPGAs, DFPs o.ä.) dadurch gekennzeichent,

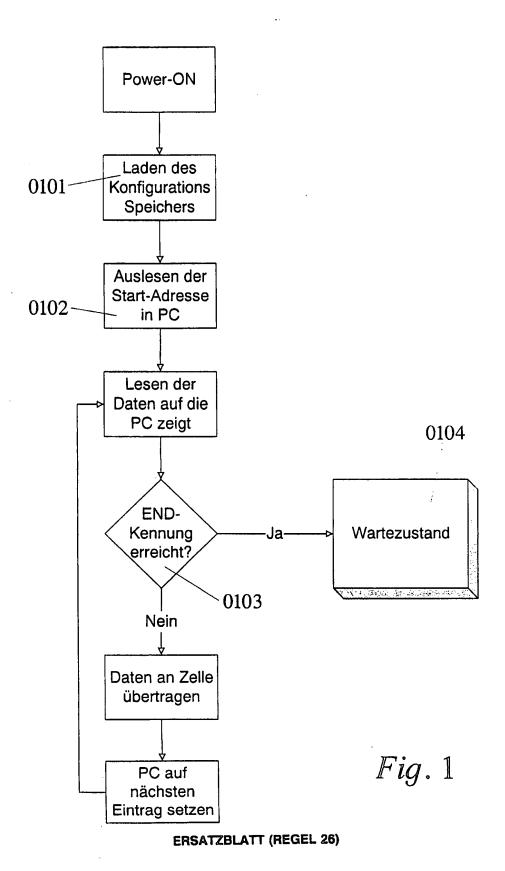
- daß 1. eine Ladelogik oder mehrere Ladelogiken
  existieren, welche auf Signale, gleich welcher Art,
  reagieren und spezielle Ladelogik-Befehle,
  innerhalb eines Konfigurationsprogramms, bestehend
  aus Daten und Befehlen, erkennen und verarbeiten
  können, sowie auf Grund der Quelle eines
  Ereignisses einen Eintrag in einer Sprung-Tabelle
  berechnen können, und
- daß 2. eine oder mehrere Sprung-Tabellen zum Auffinden der Adresse der zu ladenden Konfigurationsdaten, welche berechnet wurde, existieren,
- daß 3. ein oder mehrere Konfigurations-Speicherbereiche existieren, in denen ein oder mehrere Konfigurationsprogramme geladen werden,
- daß 4. ein oder mehrere FIFO-Speicherbereiche existieren, in den Konfigurationsdaten kopiert werden, welche nicht an die oder das zu konfigurierende Element gesandt werden konnten,
- daß 5. ein Ereignis eintrifft und auf Grund der Quelle des Ereignisses eine Adresse in einer Sprung-Tabelle berechnet wird,
- daß 6. ein FIFO-Speicherbereich, der vor jedem Umladen durchlaufen wird, und falls die Zelle nicht umgeladen werden kann, die Konfigurationsdaten näher an den Anfang des FIFO-Speicherbereichs kopiert werden, oder, falls die Zelle umgeladen werden kann, die Konfigurationsdaten an die Zelle übertragen werden,

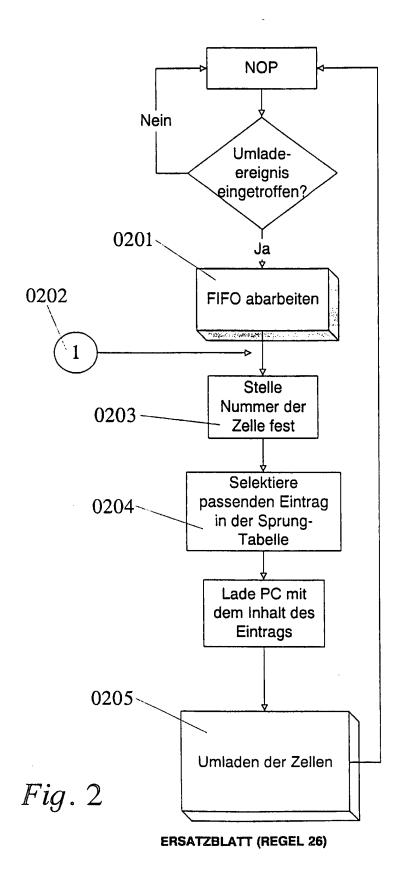
daß 8. der berechnete Sprung-Tabellen Eintrag ausgelesen wird, und die Konfigurationsdaten, welche an der ausgelesenen Adresse gespeichert sind, in die Zellen geladen werden oder, falls die Zelle nicht umprogrammiert werden kann, in den FIFO-Speicherbereich kopiert werden,

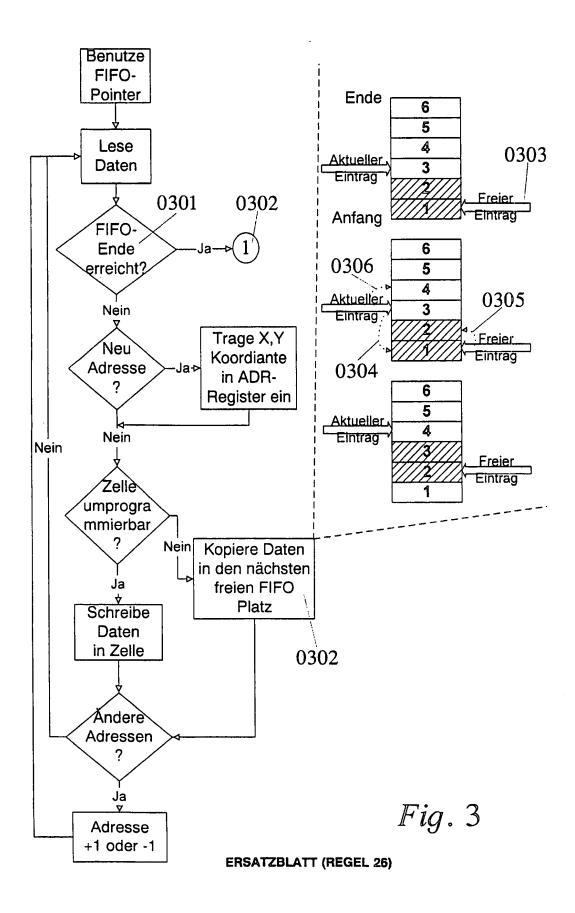
- daß 9. die Ladelogik in einen Zustand zurückspringt, in dem sie auf Ereignisse warten und auf diese reagieren kann.
- 2. Verfahren nach 'Anspruch 1, dadurch gekennzeichnet, daß der Konfigurationspeicher eine oder mehrere Konfigurationen speichert, welche eine oder mehrere komplette Konfigurationen für einen oder mehrere Bausteine enthält.
- 3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der Konfigurationspeicher eine oder mehrere Teilkonfigurationen speichert, welche nur einen Teil einer kompletten Konfiguration, eines oder mehrerer Bausteine darstellt.
- 4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Ladelogik ein Start-Konfigurations Register enthält, welches auf eine Startkonfiguration zeigt, welche den oder die Bausteine in einen gültigen Zustand versetzt.
- 5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Ladelogik ein FIFO-Start Register enthält, welches auf den Beginn des speziellen Speicherbereichs zeigt, in den Konfigurationsdaten kopiert werden.
- 6. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß

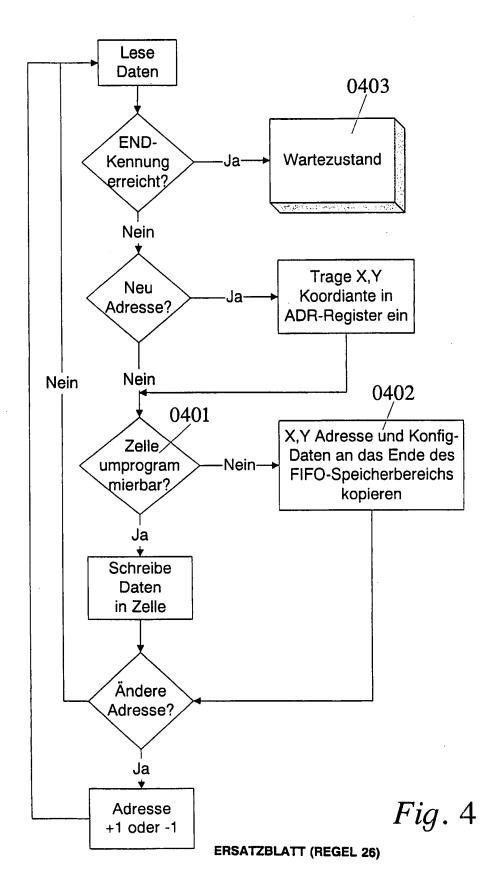
die Ladelogik ein FIFO-End Register enthält, welches auf das Ende des speziellen Speicherbereichs zeigt, in den Konfigurationsdaten kopiert werden.

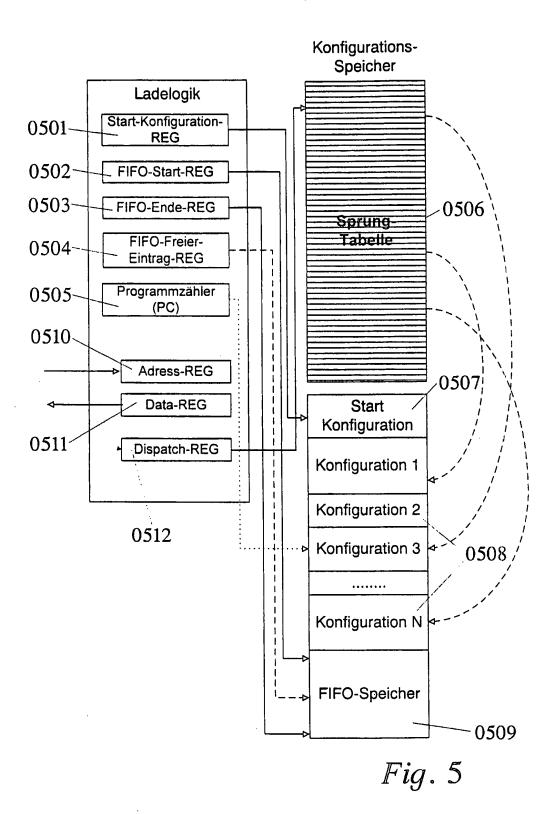
- 7. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Ladelogik ein FIFO-Free-Entry Register enthält, welches auf den ersten freien Eintrag, des speziellen Speicherbereichs zeigt, in den Konfigurationsdaten kopiert werden, zeigt, der dem Beginn dieses Speicherbereichs am nächsten ist.
- 8. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Ladelogik ein Programmzähler Register enthält, welches auf den zu verarbeitenden Eintrag innerhalb des Konfigurationsspeichers zeigt.
- 9. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Ladelogik ein Adress Register enthält, welches die Adresse (Nummer, Koordianten etc.) der Zelle enthält, welche ein Ereignis ausgelöst hat.
- 10. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Ladelogik ein Data Register enthält, welches die Konfigurationsdaten der Zelle enthält, welche an die Zelle, bei einer Umkonfigurierung, übertragen werden.
- 11. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Ladelogik ein Dispatch Register enthält, welches die aus der Zell Adresse berechnete Adresse des Eintrags in der Sprung-Tabelle enthält.











**ERSATZBLATT (REGEL 26)** 

### Konfigurationsprogramm Sprung-Tabelle 1. Befehl Adresse = 0x12161 Eintrag 39: 0x12354 0601 Eintrag 40: 0x30078 0603 1,1 ADD Eintrag 41: 0x12161 1,2 MUL 1,3 MUL 1,1 CONNECT 1,2 A 1,2 CONNECT 1,3 A Eintrag 39: 0x12354 0602 DISPATCH 41,0x34454 Eintrag 40: 0x30078 0604 **END** Eintrag 41: 0x34454 **6605** 1. Befehl Adresse = 0x3445A Eintrag 11: 0x12387 0607 1,1 DIV Eintrag 12: 0x08178 1,2 NOT 2,3 ADD Eintrag 13: 0x82161 1,1 CONNECT 1,2 A 1,2 CONNECT 2,3 A 1,1 CONNECT 2,3 B Eintrag 11: 0x12387 DISPATCH 12,0x45478 0606 Eintrag 12: 0x30178 END Eintrag 13: 0x82161 0608

Fig. 6

**ERSATZBLATT (REGEL 26)** 

## INTERNATIONAL SEARCH REPORT

Interna al Application No PCT/DE 97/02999

	·	1 ' 5 '	,
A. CLASS IPC 6	BFICATION OF SUBJECT MATTER H03K19/177 G06F17/50		
According t	to International Patent Classification(IPC) or to both national classific	ation and IPC	
	SEARCHED		
	locumentation searched (classification system followed by classificat H03K G06F	ion symbols)	
Documenta	ation searched other than minimumdocumentation to the extent that o	such documents are included in t	he fields searched
Electronic o	data base consulted during the international search (name of data ba	ase and, where practical, search	terms used)
C. DOCUM	ENTS CONSIDERED TO BE RELEVANT		
Category °	Citation of document, with indication, where appropriate, of the re-	evant passages	Relevant to claim No.
A	EP 0 678 985 A (XILINX INC) 25 0 1995 see the whole document	ctober	1
A	EP 0 748 051 A (IBM) 11 December see the whole document	1996	1
Α	MAXFIELD C: "Logic that mutates while-u-wait" EDN (EUR. ED.) (USA), EDN (EUROP EDITION), 7 NOV. 1996, CAHNERS PUSA, vol. 41, no. 23, ISSN 0012-7515, pages 137-140, 142, XP002064224 see page 138, right-hand column, page 139, left-hand column, line figures 3,4	UBLISHING, line 3 -	1
<u>ت</u>	ther documents are listed in the continuation of box C.	Patent family members	s are listed in annex.
"A" docum conside "E" earlier filling of "L" docume which citation "O" docume other "P" docume later to	ent which may throw doubts on priority claim(s) or is cited to establish the publicationdate of another on or other special reason (as specified) lent referring to an oral disclosure, use, exhibition or means ent published prior to the international filing date but han the priority date claimed	cited to understand the prinvention  "X" document of particular relectannot be considered novinvolve an inventive step to the cannot be considered to independ the cannot be considered to independ the combined with ments, such combination in the art.  "&" document member of the s.	conflict with the application but inciple or theory underlying the vance; the claimed invention el or cannot be considered to when the document is taken alone vance; the claimed invention move an inventive step when the hone or more other such docupating obvious to a person skilled ame patent family
_	actual completion of theinternational search	Date of mailing of the intere	national search report
	May 1998  mailing address of the ISA  European Patent Office, P.B. 5818 Patentlaan 2  NL - 2280 HV Rijswijk	28/05/1998 Authorized officer	
	Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Michel, T	

1

## INTERNATIONAL SEARCH REPORT

Interna al Application No PCT/DE 97/02999

Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT				
legory °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
	SALEEBA M: "A self-contained dynamically reconfigurable processor architecture" SIXTEENTH AUSTRALIAN COMPUTER SCIENCE CONFERENCE. ACSC-16, BRISBANE, QLD., AUSTRALIA, 3-5 FEB. 1993, vol. 15, no. 1, pt.A, ISSN 0157-3055, AUSTRALIAN COMPUTER SCIENCE COMMUNICATIONS, 1993, AUSTRALIA, pages 59-70, XP002064400 see the whole document	1		
	·			

### INTERNATIONAL SEARCH REPORT

Information on patent family members

Interna il Application No PCT/DE 97/02999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0678985 A	25-10-95	US 5426378 A JP 8051356 A	20-06-95 20-02-96
EP 0748051 A	11-12-96	US 5646544 A JP 8330945 A	08-07-97 13-12-96

#### Δ.

### INTERNATIONALER RECHERCHENBERICHT

Intern. Iales Aktenzeichen
PCT/DE 97/02999

		PCT/DE 9	7/02999
A. KLASSI IPK 6	ifizierung des anmeldungsgegenstandes H03K19/177 G06F17/50		
Nach der In	iternationalen Patentklassifikation (IPK) oder nach der nationalen Kla	ssifikation und der IPK	
	RCHIERTE GEBIETE		
Recherchies IPK 6	rter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbo H03K G06F	ole )	
Recherchie	rte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, so	oweit diese unter die recherchierten Gebiet	e fallen
Während de	er internationalen Recherche konsultlerte elektronische Datenbank (N	lame der Datenbank und evtl. verwendete	Suchbegriffe)
C. ALS WE	SENTLICH ANGESEHENE UNTERLAGEN	<del></del>	
Kategorie	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angab	e der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	EP 0 678 985 A (XILINX INC) 25.04 1995 siehe das ganze Dokument	ctober	1
А	EP 0 748 051 A (IBM) 11.Dezember siehe das ganze Dokument	1	
Α	MAXFIELD C: "Logic that mutates while-u-wait" EDN (EUR. ED.) (USA), EDN (EUROPE EDITION), 7 NOV. 1996, CAHNERS PUUSA, Bd. 41, Nr. 23, ISSN 0012-7515, Seiten 137-140, 142, XP002064224 siehe Seite 138, rechte Spalte, Zeile 35 Abbildungen 3,4	JBLISHING, Zeile 3 -	1
X Weite entne	ere Veröffentlichungen sind der Fortsetzung von Feld C zu ahmen	X Siehe Anhang Patentiamilie	
"A" Veröffer aber ni "E" älteres [C Anmeld "L" Veröffen acheine andere soll ode ausgef "O" Veröffer eine soll ode ausgef "P" Veröffer dem be	ntlichung, die den allgemeinen Stand der Technik definiert, icht als besonders bedeutsam anzusehen ist  Dokument, das jedoch erst am oder nach dem internationalen  dedatum veröffentlicht worden ist  tillichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft er- en zu lassen, oder durch die das Veröffentlichungsdatum einer  ni im Recherchenbericht genannten Veröffentlichung belegt werden  er die aus einem anderen besonderen Grund angegeben ist (wie  ührt)  ntlichung, die sich auf eine mündliche Offenbarung,  enutzung, eine Ausstellung oder andere Maßnahmen bezieht  mitlichung, die gine mitternationalen Anmelderdatum, aber nach	"T" Spälere Veröffentlichung, die nach der oder dem Prioritätsdatum veröffentlich Anmeldung nicht kollidiert, sondern nie Erlindung zugrundellegenden Prinzip. Theorie angegeben ist "X" Veröffentlichung von besonderer Bedekann allein aufgrund dieser Veröffentlichten von besonderer Bedekann nicht als auf erlinderischer Tätigk werden, wenn die Veröffentlichung m Veröffentlichungen dieser Kategorie is diese Verbindung für einen Fachman "&" Veröffentlichung, die Mitglied derseibe	nt worden ist und mit der ur zum Verständnie des der soder der ihr zugrundeliegenden eutung; die beanspruchte Erfindung ichung nicht als neu oder auf achtet werden uttung; die beanspruchte Erfindung keit beruhend betrachtet teiner oder mehreren anderen verbindung gebracht wird und n naheiliegend ist n Patentfamilie ist
	.Mai 1998	Absendedatum des Internationalen R 28/05/1998	ocrerene idenditis
Name und P	Postanschrift der Internationalen Recherchenbehörde Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo ni, Fax: (+31-70) 340-3016	Bevoilmächtigter Bediensteter  Michel, T	

1

## INTERNATIONALER RECHERCHENBERICHT

Intern. (ales Aktenzeichen
PCT/DE 97/02999

	PCT/DE 9		7/02999		
C.(Fortsetz	C.(Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN				
Kategorie°	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht komm	nenden Telle	Betr. Anspruch Nr.		
A	SALEEBA M: "A self-contained dynamically reconfigurable processor architecture" SIXTEENTH AUSTRALIAN COMPUTER SCIENCE CONFERENCE. ACSC-16, BRISBANE, QLD., AUSTRALIA, 3-5 FEB. 1993, Bd. 15, Nr. 1, pt.A, ISSN 0157-3055, AUSTRALIAN COMPUTER SCIENCE COMMUNICATIONS, 1993, AUSTRALIA, Seiten 59-70, XP002064400 siehe das ganze Dokument		1		

#### .C . C = GA

# INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Interna ales Aktenzeichen
PCT/DE 97/02999

lm Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
EP 0678985 A	25-10-95	US 5426378 A JP 8051356 A	20-06-95 20-02-96
EP 0748051 A	11-12-96	US 5646544 A JP 8330945 A	08-07-97 13-12-96